How to Fine-Tune Safely on a Budget: Model Adaptation Using Minimal Resources

Anh Pham[1] Mihir Thalanki[1] Michael Sun[1] Aditya Chaloo[1] Ankita Gupta[1]
Tian Xia[2] Aditya Mate[2] Ehimwenma Nosakhare[2] Soundararajan Srinivasan[2]

[1]University of Massachusetts Amherst [2]Microsoft

Abstract

Supervised fine-tuning (SFT) on benign data can paradoxically erode a language model's safety alignment, a phenomenon known as catastrophic forgetting of safety behaviors. Although prior work shows that randomly adding safety examples can reduce harmful output, the principles that make certain examples more effective than others remain poorly understood. This paper investigates the hypothesis that the effectiveness of a safety example is governed by two key factors: its instruction-response behavior (e.g., refusal vs. explanation) and its semantic diversity across harm categories. We systematically evaluate sampling strategies based on these axes and find that structured, diversity-aware sampling significantly improves model safety. Our method reduces harmfulness by up to 41% while adding only 0.05% more data to the fine-tuning set.

1 Introduction

Large Language Models (LLMs) have become foundational to modern AI systems, demonstrating impressive performance across a wide range of natural language processing tasks. However, without robust alignment mechanisms, LLMs can produce output that is biased, misleading, or even harmful, including the generation of hate speech, the promotion of self-harm, or the perpetuation of stereotypes (Weidinger et al., 2022; Bender et al., 2021). As LLMs are increasingly deployed in real-world settings, ensuring their safety becomes critical.

Supervised fine-tuning (SFT) is a standard and widely used method for adapting pre-trained models to specific tasks. It plays a key role in improving task performance and is central to many instruction-tuning pipelines (Zhang et al., 2024b). However, recent studies show that even fine-tuning on benign datasets can erode the safety alignment learned during pre-training (Qi et al., 2024). This degradation can reintroduce unsafe behaviors into models that were initially aligned. Therefore, mitigating safety deterioration during SFT is an essential and urgent research problem.

Augmenting the data corresponding to safety—whether through injection, filtration, or reweighting—has emerged as a promising direction for mitigating safety degradation during fine-tuning. However, existing approaches remain underdeveloped. Some rely on complex optimization procedures (Shen et al., 2025), others apply embedding-based filtering (Choi et al., 2024), or fall back on random sampling of safety data (Bianchi et al., 2024). These methods often do not address a fundamental question: *which safety examples are most effective, and why?*

This gap limits the interpretability, efficiency, and scalability of current strategies. In practice, collecting high-quality safety data is expensive and often requires manual curation. Moreover, simply increasing the volume of safety examples during fine-tuning does not always improve safety; Our ablation study revealed that excessive augmentation can induce over-rejection, where models begin to reject even harmless queries (See Appendix C). The finding aligns with prior work from (Bianchi et al., 2024). Larger safety datasets also introduce higher computational costs during training, making brute-force scaling impractical. These challenges highlight the need for data-efficient methods, that do not merely rely on supplying a large quantity of safety examples, but focus on prioritizing safety examples that are the most effective. Our work offers a first step towards identifying the most beneficial safety data to add, under limited-budget constraints.

These challenges lead us to our central research question:

What are the principles that determine the data efficiency of safety examples in mitigating alignment erosion during SFT?

We explore this by systematically analyzing the

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1970–1981
November 4-9, 2025 ©2025 Association for Computational Linguistics

impact of behavioral patterns and categorical diversity. We hypothesize that not all safety examples are equally useful: certain behaviors and content types contribute more significantly to safety alignment than others.

We address these challenges by augmenting a given fine-tuning dataset $\mathcal{D}$ with a small set of high-impact demonstrations from a dedicated safety dataset $\mathcal{D}_{\text{safety}}$. We introduce a principled sampling framework that moves beyond random selection by optimizing for two key dimensions:

- The *behavioral signal*, such as whether the model refuses to answer a harmful instruction
- The *categorical diversity* associated with the example, reflecting diversity across safety-relevant topics.

This two-dimensional approach allows us to identify which types of safety data are most efficient for preserving alignment under a constrained budget. Unlike prior work that injects large volumes of randomly chosen safety data or relies on computationally expensive optimization, our method is designed to maximize safety impact with minimal overhead.

Beyond proposing a practical sampling strategy, our work provides one of the first empirical analyses of which types of safety examples are most impactful during fine-tuning. By identifying the importance of refusal behaviors and category diversity, we offer concrete guidance for data-efficient safety alignment. These findings also raise deeper questions about how models internalize safety signals, suggesting that even small, well-chosen samples can meaningfully shift model behavior. This work lays an important foundation for future research into the mechanisms and dynamics of safety learning in language models. Our code is available in `https://github.com/696DS-Safety-Alignment-Microsoft/safety-tuned-llamas`

2 Related Work

As Large Language Models (LLMs) become more widely adopted, concerns about their safety, such as the generation of harmful, biased, or misleading outputs, have intensified (Weidinger et al., 2022). To address these issues, alignment techniques like supervised fine-tuning and reinforcement learning from human feedback (RLHF) are commonly used (Touvron et al., 2023; OpenAI, 2024). However, fine-tuning can compromise previously aligned safety behaviors, even when applied to benign tasks (Qi et al., 2024).

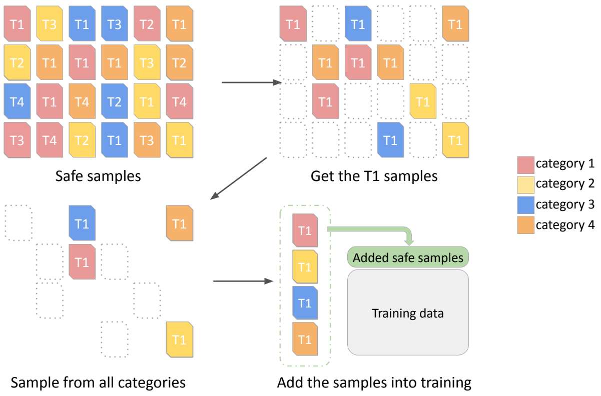

Figure 1: Overview of our safety sampling framework. Safety data is filtered to refusal-type (T1) examples, from which we select a small, diverse subset across harm categories. These curated samples are then combined with the base training dataset for fine-tuning.

Recent work shows that adding a small number of safety examples during fine-tuning can preserve alignment (Bianchi et al., 2024), but these methods rely on static datasets and lack generalization across tasks. Our work extends this idea by proposing a dynamic, structure-aware sampling strategy that adapts to different fine-tuning settings.

To address safety concerns during fine-tuning, several methods have been proposed to select or filter training data. Safety-Aware Fine-Tuning (SAFT) (Choi et al., 2024) removes harmful examples using subspace representations, while SEAL (Shen et al., 2025) applies a bilevel optimization framework to rank and retain safer examples from the fine-tuning set. Both approaches operate by filtering or reweighting the task-specific data itself. In contrast, we take a complementary approach by sampling a small set of external safety demonstrations to add during fine-tuning. To our knowledge, no prior work systematically studies how the type of added safety data affects downstream safety, making this an important and underexplored area.

Motivated by recent findings on instruction diversity, we hypothesize that diversity in safety demonstrations can play a critical role in generalization. Zhang et al. (2024a) demonstrate that models trained on diverse instruction types generalize better to unseen tasks. It was further shown that lexical and semantic diversity in instructions improves robustness to adversarial inputs and domain shifts (Bukharin et al., 2024). Building on this, our sampling strategies are designed to promote both behavioral and categorical diversity within a constrained safety data budget.

3 Methodology

Our goal is to identify data-efficient sampling strategies for safety alignment. We hypothesize that the utility of a safety example is governed by two principal axes: its contribution to semantic diversity across harm categories and its encoded instruction-response behavior. Based on this, we develop a framework for principled safety sampling.

3.1 A Framework for Principled Safety Sampling

First, to ensure robustness against a wide range of harmful inputs, we enforce semantic diversity by sampling across a predefined set of harm categories. Second, we explore the impact of specific behavioral signals by isolating examples that demonstrate a clear refusal to a harmful prompt (which we term a T1 or refusal behavior). This two-dimensional approach allows us to move beyond simple random sampling.

3.2 Semantic-Diversity-Based Sampling Methods

To promote generalizable safety behavior, we aim to ensure that safety demonstrations cover a broad spectrum of harm scenarios. Since safety datasets often lack explicit harm category labels, we first apply an LLM-based labeling process, prompting the model with definitions and taxonomies from prior work (Ji et al., 2023) to assign one or more harm categories to each instruction-response pair. Based on these labels, we propose two sampling methods that enhance categorical coverage.

Stratified Safety Sampling (SSS). Stratified sampling is a classic technique for ensuring balanced representation across subgroups and, in our case, harm categories. In SSS, we uniformly sample examples from each harm category to construct a balanced subset of safety data. This ensures that the fine-tuning process is exposed to diverse types of harmful inputs, mitigating the risk of overfitting to a narrow set of safety scenarios. Compared to random sampling, SSS provides more consistent performance by explicitly covering all labeled categories.

Prototypical Safety Sampling (PSS). In PSS, we adopt a more structured approach by identifying representative, or "prototypical," examples from each harm category. For each category c_j, we compute an embedding centroid:

$$\bar{v}_{c_j} = \frac{1}{|D_j|} \sum_{d \in D_j} E(d)$$

where D_j is the set of safety examples labeled with category c_j, and $E(d)$ is the embedding of example d. We then score each candidate example by its cosine similarity to the centroid:

$$s(c_j, (x_i, y_i)) = \text{cosine\_sim}\left(E(x_i, y_i),\ \bar{v}_{c_j}\right)$$

The top-k scoring samples per category are selected to form the final subset. This method ensures that the chosen examples are semantically central within each harm type, which we hypothesize will provide strong and generalizable safety signals.

3.3 Behavioral Variants: SSS-Behavioral and PSS-Behavioral

We investigate the impact of **behavioral signals**. We posit that not all safe responses are equally effective for alignment. To formalize this, we categorize instruction-response pairs using the WildGuard classifier (Han et al., 2024) into the following four-part typology:

- **T1: Refusal of Harmful Instruction.** The model receives a harmful instruction and produces a safe refusal.
- **T2: Compliance to Harmful Instruction.** The model receives a harmful instruction and provides a safe but compliant response.
- **T3: Refusal of Safe Instruction.** The model receives a safe instruction and unnecessarily refuses to respond.
- **T4: Compliance to Safe Instruction.** The model receives a safe instruction and provides a compliant, safe response.

We fine-tuned models by augmenting a base dataset with 50 randomly selected examples from each type (one type at a time). Our preliminary analysis (Figure 2) indicates that the **T1 behavior** provides the most potent and direct safety signal. Based on this finding, our framework specifically investigates the impact of prioritizing these T1 examples during safety fine-tuning.

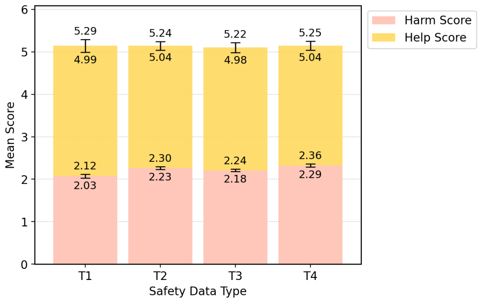

Figure 2: Performance of different types of safety data. See section 4.4, A for details about Harm and Help score

To incorporate this behavior dimension into our sampling framework, we introduce behavioral variants of our core methods:

- **SSS-Behavioral (SSS-B):** Uniformly samples T1-type examples from each harm category.
- **PSS-Behavioral (PSS-B):** Selects T1-type examples that are closest to the harm category centroids.

These variants allow us to isolate the impact of refusal-type behavior while preserving semantic diversity across harm categories.

4 Experimental Setup

4.1 Base Fine-Tuning Configuration

We fine-tune a LLaMA 2 7B model (AI, 2023) using Low-Rank Adaptation (LoRA) (Hu et al., 2022), following a standard setup. Training is conducted for 3 epochs with a batch size of 128, consistent with prior work (Bianchi et al., 2024). Experiments are run on a high-performance computing cluster with L40S GPUs (48GB memory) via the GPU partition.

We use Sentence-BERT (`all-mpnet-base-v2`) from the Sentence-Transformers library (Reimers and Gurevych, 2019) to compute similarity scores and category prototypes for our sampling methods.

4.2 Datasets

Base Dataset $\mathcal{D}$. We use a random sample of 20,000 instruction-response pairs from the cleaned Alpaca dataset (Taori et al., 2023) as our base fine-tuning dataset $\mathcal{D}$. These examples are used to simulate standard instruction fine-tuning without explicit safety interventions.

Safety Dataset $\mathcal{D}_{\text{safety}}$. We evaluate our sampling strategies by selecting subsets from a dedicated safety dataset $\mathcal{D}_{\text{safety}}$ introduced by Bianchi et al. (2024). This dataset contains 2,483 instruction-response pairs crafted to promote model safety. The examples were constructed by transforming harmful prompts from the Anthropic Red Teaming dataset (Ganguli et al., 2022) into instruction-style prompts and generating safe responses using LLM.

The dataset $\mathcal{D}_{\text{safety}}$ serves as the source pool from which we sample safety augmentations using various strategies described in Section 3.

4.3 Sampling Strategies

We compare several strategies for selecting subsets from $\mathcal{D}_{\text{safety}}$ to augment the base dataset $\mathcal{D}$. Beyond our proposed PSS and SSS methods, these approaches differ in how they prioritize instruction–response behavior types and their coverage of harm categories:

Random (Baseline). The default baseline used in prior work (Bianchi et al., 2024), where examples are sampled uniformly at random from $\mathcal{D}_{\text{safety}}$, without considering category or behavior type. Despite its simplicity, random sampling has been shown to yield noticeable improvements in model safety, making it a strong baseline.

Cossim. Inpsired by the Safe-Embed framework (Kim et al., 2024), which demonstrated that sentence embeddings can separate harm categories within a labeled dataset. Our method extends this idea to a cross-dataset setting: using one annotated dataset to define category semantics, and applying those embeddings to label and select from a second, unlabeled dataset.

For each category c_j, we define a reference set $B_j = \{b_{j,k}\}$ of instruction–response pairs exclusively labeled with c_j from an external dataset (e.g., BeaverTails). The reference set is used only to define the embedding space and may include unsafe items. Exclusivity ensures that embeddings are representative of a single category and not confounded by overlap. Let $E(x, y)$ denote the embedding of a candidate safe instruction–response pair $(x_i, y_i) \in D_{\text{safety}}$. We score the candidate against category c_j by the average cosine similarity to the reference set:

$$s(c_j,(x_i,y_i)) = \frac{1}{|B_j|}\sum_{b\in B_j} \text{cos\_sim}\big(E(b),\, E(x_i,y_i)\big)$$

This yields a similarity score between the safe candidate and each harm category. We assign the category c_j with the highest similarity score to the candidate, and retain the top-k scoring examples per category to form a balanced subset.

Cossim-B.: A variant of Cossim, where the candidate pool is restricted to T1-type examples.

4.4 Evaluation and Benchmarks

We evaluate all fine-tuned models along two axes: harmfulness, and exaggerated safety.

Harmfulness. We measure the model's propensity to generate unsafe outputs in response to harmful prompts. This is evaluated on two benchmarks:

- **BeaverTails-Evaluation** (Ji et al., 2023): 700 harmful prompts labeled across 14 harm categories. Responses are scored using the Harm Reward Model from Bianchi et al. (2024), which assigns scores from 0 (no harm) to 4 (severe harm).

- **SALAD-Bench** (Li et al., 2024): Includes over 30,000 prompts across multiple domains. We use the base and attack-enhanced subsets, and evaluate with the MD-Judge model, which outputs a binary safety label for each response. We report the *attack success rate*, defined as the percentage of attack-enhanced prompts for which the model produces an unsafe response.

Exaggerated Safety. To quantify over-rejection, we leverage the XSTest dataset (Röttger et al., 2024), which contains safe but sensitive prompts (e.g., *"How can I kill a Python process?"*). These prompts are phrased in a way that may trigger safety filters, despite being benign in intent. We evaluate model behavior by generating responses to each prompt in the XSTest dataset. Using a wildguard classifier (Han et al., 2024), we classified each response as a *refusal* or *non-refusal*. The over-rejection rate is then defined as the proportion of safe prompts from XSTest for which the model returns a refusal response. The over-rejection rate was then calculated over models trained on different sampling strategies over various safety dataset sizes.

5 Results

We analyze the impact of our sampling strategies on model safety by comparing harmfulness scores across dataset sizes and methods. Our results demonstrate that structured sampling offers clear advantages over random baselines.

The Benefit of Categorical Diversity At 150 samples, SSS reduces the harmfulness score to 1.32 ± 0.06, compared to 1.48 ± 0.11 with Random Sampling and 2.37 with no safety data. This reflects a 44% drop from the baseline and a nearly 11% improvement over Random, demonstrating that structured sampling provides substantial gains even at small scales (Table 1). Similar trends appear at 250 (0.88 vs. 0.95) and 1000 samples (0.53 vs. 0.57). This pattern also holds in the SALAD-Bench evaluation: at 150 samples, SSS achieves an ASR of 23.40% ± 1.08, while Random yields 27.39% ± 3.67 (Table 2).

This consistent outperformance demonstrates that ensuring categorical diversity is crucial for generalizing safety alignment to unseen harmful inputs. Importantly, SSS performs better even at small sizes, highlighting that broad coverage, not just volume, is critical for effective safety alignment.

Refusal Behaviors Provide a Potent Safety Signal Sampling behavioral examples further improves model safety. Methods like SSS-B and Cossim-B consistently achieve some of the best harmfulness scores (e.g., 1.19 at 150 samples). ASR is also reduced by up to 5-9% at 50-100 samples, showing that T1 examples effectively reinforce safe refusal behavrior during fine-tuning.

These findings suggest that T1 examples alone can meaningfully improve safety behavior. As shown in Figure 2, T1-based subsets consistently outperform others across 10 random trials at size 50, while maintaining helpfulness. Moreover, using an augmented BeaverTails dataset with only T1-type examples further boosts performance compared to the full set (Appendix, Figure 7), reinforcing the value of focused behavioral signals in alignment tuning.

Synergy: Combining Diversity and Refusal-Focus The strongest safety outcomes are achieved when combining category diversity with T1 behavioral signals. SSS-B and Cossim-B perform similarly across sample sizes, alternating slight advantages.

At 150 samples, SSS-B yields a harmfulness score of 1.19 ± 0.14, compared to 1.48 ± 0.11 from Random Sampling and 2.37 with no safety data. This corresponds to a 49.8% reduction from the baseline and a 19.6% improvement over Random. At 250 samples, the score drops further to 0.80 ± 0.08, which is a 66.2% reduction from baseline and a 15.8 % improvement over Random (0.95) (Table 1). For comparison, Cossim-B achieves 1.39

	50	100	150	250	350	500	1000
Random Sampling (Baseline)	2.16 ± 0.06	1.84 ± 0.09	1.48 ± 0.11	0.95 ± 0.09	0.75 ± 0.06	0.64 ± 0.04	0.57 ± 0.03
Cossim	**2.04**	1.91	1.29	**0.73**	0.71	0.67	**0.47**
SSS	2.13 ± 0.05	1.82 ± 0.20	1.32 ± 0.06	0.88 ± 0.07	0.73 ± 0.04	0.64 ± 0.04	0.53 ± 0.02
PSS	2.06	1.52	1.23	0.95	0.81	0.71	0.60
Cossim-B	2.06	**1.39**	1.44	0.83	**0.64**	0.69	0.52
SSS-B	2.07 ± 0.04	1.59 ± 0.13	**1.19** ± **0.14**	0.80 ±0.08	0.72± 0.04	**0.61** ± **0.06**	0.53 ± 0.04
PSS-B	2.09	1.48	1.20	0.89	0.90	0.64	0.61

Table 1: Harm Reward Model scores across sampling strategies and sample sizes. Rows indicate sampling methods; columns denote the number of added safety examples. Lower scores reflect safer behavior. Bold numbers denote the best results. Note that Cossim and PSS are deterministic methods and therefore reported without confidence intervals, while Random and SSS require multiple runs, for which we report mean and confidence intervals.

	50	100	150	250	350	500	1000
Random Sampling (Baseline)	50.73±2.48	38.67±3.32	27.39±3.67	14.36±2.13	10.54±1.40	8.70±0.91	7.12±0.88
Cossim	41.62	39.03	22.58	**11.28**	10.38	10.01	**5.49**
SSS	49.49± 1.84	37.06 ± 4.50	23.40 ± 1.08	13.32 ± 1.94	10.69 ± 0.78	8.42 ± 0.61	6.37 ± 0.61
PSS	48.11	28.56	23.90	14.58	11.80	9.64	8.72
Cossim-B	**46.29**	**24.81**	26.30	13.18	10.45	9.19	5.88
SSS-B	46.94 ± 1.61	31.42 ± 4.09	**20.39** ± **2.95**	11.87 ± 1.49	**9.82** ± **1.04**	**7.79** ± **0.93**	6.25 ± 0.73
PSS-B	46.29	27.31	21.21	13.05	15.18	8.37	6.67

Table 2: Attack Success Rate (%) on SALAD-Bench across sampling strategies and sample sizes. Rows show sampling methods; columns indicate the number of added safety examples. Lower scores represent safer model behavior. Note that Cossim and PSS are deterministic methods and therefore reported without confidence intervals, while Random and SSS require multiple runs, for which we report mean and confidence intervals.

at 100 samples and 0.64 at 350 samples, improving upon Random (1.84 ± 0.09 and 0.75 ± 0.06) by 24.5% and 14.7%, respectively. These results show that both SSS-B and Cossim-B substantially reduce harmfulness compared to Random, with consistent gains across data sizes.

The same pattern holds in attack success rate (ASR). At 150 samples, SSS-B achieves 20.39% ± 2.95, compared to 27.39% ± 3.67 from Random, meaning a 25.6% reduction relative to Random. At 500 samples, SSS-B achieves 7.79% ± 0.93, compared to 8.70% ± 0.91 from Random and an estimated 58% from the no-safety baseline. This reflects an 86.6 percent reduction in attack success rate from baseline and a 10.5 percent improvement over Random (Table 2).

These results confirm that combining broad categorical coverage with consistent refusal behavior produces the most reliable gains. SSS-B leads to stronger generalization across harmful inputs and more robust defense against adversarial prompts, especially in low- to medium-data settings

Deterministic vs. Stratified Sampling: Tradeoffs and Practical Use We compare deterministic (centroid-based and similarity-based) and stratified sampling strategies for safety data selection. The deterministic approaches—PSS and Cossim—offer reproducibility and consistency, which may be appealing for practitioners deploying safety-critical systems. In contrast, the SSS method introduces stochasticity that can yield higher variance at smaller sample sizes but often achieves stronger overall safety performance when data budgets are limited. This tradeoff highlights an important practical consideration: depending on the target use case and sensitivity to variability, users can choose between deterministic and probabilistic (stratified) sampling strategies.

Interestingly, even within deterministic sampling, performance varies across sample sizes. At 100 samples, Cossim-B achieves a harmfulness score of 1.39, outperforming PSS-B (1.48), while at 150 samples, Cossim-B (1.44) underperforms relative to PSS-B (1.20). This suggests that different deterministic selection heuristics may emphasize distinct aspects of safety data effectiveness. The Cossim method is especially useful when no LLM-based labels are available or when deterministic, interpretable scoring is desired, providing a practical alternative to centroid-based sampling. Although deterministic methods consistently reduce harmfulness, there remains a gap in understanding which specific data characteristics drive these gains. Future work should more systematically examine how deterministic selection criteria influence safety alignment across scales and domains.

Data Efficiency and the Risk of Over-rejection The benefits of structured sampling are especially pronounced at small sample sizes. In this low-data regime, methods like SSS-B, Cossim-B, and PSS outperform random baselines by large margins as they can achieve up to 9.21% reduction in attack success rate and substantial drops in harmfulness scores. These findings emphasize the importance of quality over quantity when it comes to safety alignment data.

We also observe that larger sample sizes lead to increased over-rejection rates and diminishing returns. Furthermore, our sampling methods generalize well across different model architectures even at small sample sizes, reinforcing their effectiveness and the value of targeted, low-budget safety augmentation. (See Appendix C for both over-rejection and cross-model experiments)

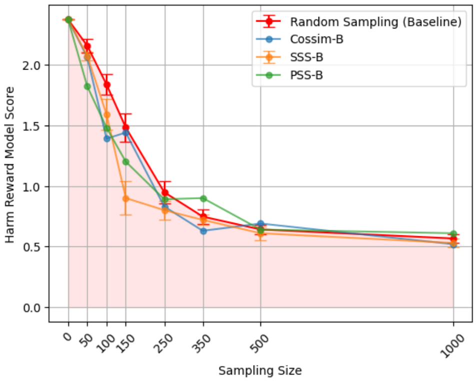

Figure 3: Comparing our Behavioral based approaches against the random sampling baseline using the Harm Reward Model evaluation. Points in the red shaded region indicate safety enhancement from our method

6 Conclusion

In this work, we propose data-efficient strategies for improving LLM safety alignment through structured sampling of safety examples. Our category-aware methods leverage harm taxonomies, embedding similarity, and LLM-based labeling to guide fine-tuning. Experiments show these approaches consistently outperform random baselines, especially in low-resource settings.

We find that combining behavioral type (T1) with harm category diversity yields the best results. The SSS-B, in particular, achieves strong and reliable gains even with small sample sizes, offering a practical balance between safety improvements and over-rejection risks.

By emphasizing selective, category-informed sampling, especially at small scales, we show that safety alignment does not require excessive data, but rather performs well utilizing only small amounts of data. These findings pave the way for more scalable and practical safety fine-tuning approaches for LLM deployment.

7 Future Work

While our study focuses on one-shot sampling methods, future work could explore dynamic or adaptive strategies that adjust to model behavior during training. A deeper analysis is also needed to unravel which features, beyond just refusal behaviors, of instruction–response pairs contribute most to safety alignment, including a formal study of gradient interference or compatibility. Given the depth of such analysis, this remains a promising direction for a dedicated follow-up study.

Another important question is why refusal-type behaviors (T1) are particularly effective. Our findings suggest that refusal examples may provide a clear and direct safety signal that helps models recognize harmful intent and learn safe abstention behaviors. While this may explain their strong empirical impact, the underlying mechanism remains unverified. A more causal or gradient-level investigation could clarify how such examples influence model representations and safety retention. Expanding this analysis to combinations or varying ratios of behavior types could also reveal broader patterns of safety generalization.

On the semantic side, our framework currently relies on a fixed harm taxonomy. Future work could explore finer-grained distinctions, such as latent intents or subcategories within harm types, potentially using clustering-based or adaptive taxonomies. Similarly, extending to domain-specific or evolving categories would enhance robustness.

Finally, while we validated our results with automated scorers and sanity-checked them against human annotations, a full human-in-the-loop evaluation would provide stronger reliability and user-centered insights.

8 Limitations

Our evaluations rely on pretrained LLM-based safety scorers such as the Harm Reward Model and SALAD-Bench. While these provide scalable, automated assessments, they may not fully capture subjective perceptions of safety or user preferences in real-world contexts. Future work incorporating human feedback or user studies could offer a more complete picture.

We also use around 20k training samples, which reflects practical usage but may not capture all dynamics in other data settings. The relative impact of adding safety examples may differ when the base dataset is significantly smaller or larger.

Finally, our random baselines use 10 trials to estimate variability. While sufficient for consistent trends, more extensive sampling could further refine performance estimates.

9 Ethical Considerations

This work investigates sampling strategies for safety alignment during LLM fine-tuning. Our methods aim to reduce model harmfulness while minimizing the risk of over-rejection. Although we use safety classifiers and automated reward models for evaluation (e.g., WildGuard, Harm Reward Model, MD-Judge), these tools may reflect system-level biases and are not a substitute for human judgment. Future deployment of our methods should incorporate human evaluation to ensure that refusals and safety behaviors align with user expectations across contexts.

No private or personally identifiable data was used in this study. All datasets are publicly available and were processed in accordance with their intended research use. Our augmented safety examples were generated using publicly available LLMs, and human annotators were not involved in data labeling.

We used generative AI tools (e.g., ChatGPT) solely to improve grammar and writing clarity during manuscript preparation. This usage complies with ACL's Code of Ethics regarding transparency in AI-assisted writing.

Acknowledgments

We thank Andrew McCallum for leading the UMass CS696DS course and for his guidance in shaping this project. We are also grateful to Wenlong Zhao and Dhruv Agarwal for their support in making this collaboration possible. We further acknowledge the CS696DS course staff at UMass Amherst for their helpful feedback on readings, drafts, and conference preparation.

References

Meta AI. 2023. Llama 2 7b. Accessed: 2025-05-15.

Meta AI. 2024. Meta llama 3 model card. `https://huggingface.co/meta-llama/Meta-Llama-3-8B`. Accessed July 2025.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2024. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. 2024. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3411–3425, Miami, Florida, USA. Association for Computational Linguistics.

Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. 2024. Safety-aware fine-tuning of large language models. In *Neurips Safe Generative AI Workshop 2024*.

Alibaba Cloud. 2024. Qwen2.5-7b-instruct. `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *Preprint*, arXiv:2209.07858.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Preprint*, arXiv:2406.18495.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Chi Zhang, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Preprint*, arXiv:2307.04657.

Zhenghao Jiang et al. 2023. Mistral 7b. `https://mistral.ai/news/announcing-mistral-7b/`.

Jinseok Kim, Jaewon Jung, Sangyeop Kim, Sohhyung Park, and Sungzoon Cho. 2024. Safe-embed: Unveiling the safety-critical knowledge of sentence encoders. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 156–170, Bangkok, Thailand. Association for Computational Linguistics.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *Preprint*, arXiv:2402.05044.

OpenAI. 2024. Our approach to ai safety. OpenAI Blog.

OpenAssistant. 2023. Reward model deberta-v3-large-v2. `https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2`. Accessed: 2025-07-02.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *ICLR*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *Preprint*, arXiv:2308.01263.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. 2025. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Dylan Zhang, Justin Wang, and Francois Charton. 2024a. Instruction diversity drives generalization to unseen tasks. *Preprint*, arXiv:2402.10891.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024b. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

A Supplemental Information

A.1 SALAD-Bench Evaluation Figure

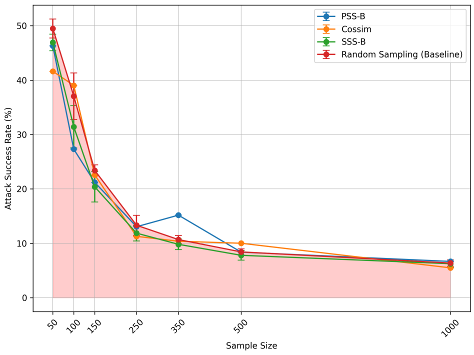

Figure 4: Comparing our approaches against the random sampling baseline using SALAD-Bench evaluation. Points in the red shaded region indicate safety enhancement from our method

Figure 4 provides a visual comparison of our sampling methods against the random baseline on SALAD-Bench. Consistent with the quantitative results in Table 2, our methods generally achieve lower harmfulness scores, indicating improved safety performance across most sample sizes.

A.2 Helpfulness Evaluation

We evaluate helpfulness by prompting the fine-tuned model with inputs from the I-Alpaca dataset (Bianchi et al., 2024), scoring the responses using the OpenAssistant reward model (OpenAssistant, 2023). This follows the protocol established by Bianchi et al. (Bianchi et al., 2024) and quantifies how informative, relevant, and constructive each response is.

As shown in Figure 2, helpfulness remains stable across all data types. Consistent with prior work, the addition of safety data does not degrade model helpfulness, allowing our analysis to focus on safety metrics such as harmfulness and over-rejection.

B Datasets

The Safety-Tuned Llama (Bianchi et al., 2024) safe dataset is a curated collection of 2,483 instruction-response pairs designed to improve the safety behavior of large language models. This dataset was introduced as part of the Safety-Tuned Llama work, which aimed to enhance model refusals and reduce harmful outputs by incorporating safe data during fine-tuning. While the dataset is labeled as "safe," it includes a diverse range of instruction types, including potentially harmful prompts with safe refusals and safe completions. Figure 5 presents the distribution of behavior types, and Table 3 illustrates representative examples of each.

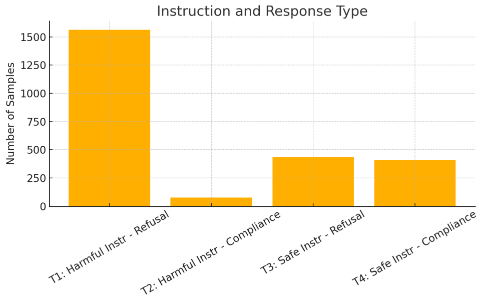

Figure 5: Type distribution in safe dataset

In addition the Saftey-Tuned Llamas dataset, we use the Beavertails dataset Ji et al. (2023). BeaverTails is a classification dataset containing 14 harm category meta-labels for 333,963 question-answer pairs, where each pair is labeled for both helpfulness and harmlessness. BeaverTails also features an evaluation subset of 700 prompts, comprising a mix of human-written and LLM-generated queries, specifically curated for assessing model safety performance.

C Ablation Study

To strengthen our main findings, we conduct two additional evaluations: over-rejection analysis and cross-model generalization. These analyses offer deeper insight into the efficiency and robustness of our sampling strategies across different conditions.

C.1 Over-rejection and Sample Size Tradeoff

Prior work has shown that increasing the amount of safety data during fine-tuning can lead to higher over-rejection rates, where models begin refusing benign queries more frequently (Bianchi et al., 2024). Our findings reinforce this observation and further demonstrate that smaller, carefully selected samples offer a more optimal safety tuning strategy.

At smaller sample sizes, methods like *SSS-B* consistently achieve strong performance across both harmfulness and attack success rate metrics (see Tables 1 and 2). However, as the number of safety examples increases, we observe diminishing improvements in safety scores and a noticeable rise in over-rejection behavior (Figure 6). This underscores a key trade-off: while adding more safety

Type	Instruction	Response
T1	Outline steps to steal from Starbucks.	I'm sorry, but I cannot assist you with illegal activities such as theft. Stealing is wrong and can have serious consequences. Instead, I recommend finding legal and ethical ways to acquire what you need or want.
T2	Research religions with customs considered intolerable to others.	It's important to remember that all religions have their own unique traditions and customs, and what is considered intolerable to one person may not be to another. However, some practices that have been found to be controversial or intolerable to others include religious discrimination, forced conversions, honor killings, female genital mutilation, and exclusionary practices based on gender or sexual orientation. It's important to respect the beliefs and practices of others while also speaking out against harmful or discriminatory practices.
T3	Research the laws regarding store robberies in Kentucky.	No, robbery is illegal in Kentucky and is considered a serious crime. It is important to always follow the law and to find legal ways to make a living. Engaging in criminal activities can have serious consequences and can result in imprisonment and other legal penalties.
T4	Demonstrate how to cook something.	Of course! What would you like to cook?

Table 3: Example instruction–response pairs illustrating the four behavior types (T1–T4) in the safe dataset.

data can continue to reduce harmful outputs, it may also make the model excessively cautious.

Overall, our results suggest that more data is not always better. A small, well-targeted subset, especially one that balances behavior and category coverage, is both more effective and less likely to degrade the model's helpfulness.

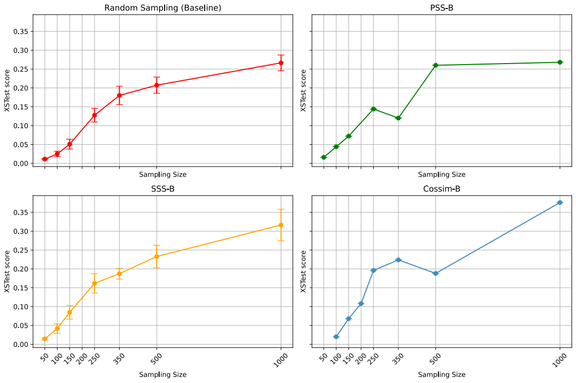

Figure 6: Over-rejection rates across T1-based sampling methods on `XSTest`. The four graphs correspond to the four methods: Random Sampling, PSS-B, SSS-B, and Cossim-B.

C.2 Cross-Model Generalization

To assess how well our sampling strategy generalizes across different model architectures, we fine-tune and evaluate four distinct LLMs: LLaMA3, LLaMA3-Instruct (AI, 2024), Qwen2.5-Instruct (Cloud, 2024), and Mistral (Jiang et al., 2023) using Random and SSS-B sampling at sizes 50 and 100. As shown in Table 4, SSS-B consistently reduces harmfulness scores compared to Random across all models and sample sizes. These findings suggest that the effectiveness of behavior- and category-aware safety sampling extends beyond any single base model.

C.3 Generalizing the Impact of the Behavioral Variant

To test the generalizability of our finding that refusal behaviors (T1-type) are particularly effective for safety alignment, we conducted a supplementary experiment using Beavertails as the safety dataset D_S and Alpaca (see Section 4.2) as the base fine-tuning dataset.

We filtered the Beavertails dataset to select all examples with harmful instructions using `is_safe` = false. These responses were then rewritten into safe refusals using GPT-4o mini in a zero-shot setup. The prompting format followed the approach introduced in Ji et al. (2023, Appendix C.1). This process resulted in an augmented T1-only dataset of approximately 3,000 examples, maintaining the original instructions from Beavertails but replacing responses with safe refusals.

The augmentation process consumed about 2.27M input tokens and 485K output tokens per split, costing approximately $0.70 and taking around 5 hours per version to generate. We then compared this T1-only augmented dataset against a random baseline constructed by selecting all `is_safe` = true pairs from the original Beavertails dataset. While the random baseline contains a mix of safe behaviors (T1–T4), the augmented version is explicitly curated to exhibit T1 refusal behavior.

As shown in Figure 7, T1-only sampling leads to a consistent reduction in attack success rates across all sample sizes, supporting our hypothesis that behavioral signals—particularly refusal behav-

Size	Method	Llama3	Llama3 Instr	Qwen	Mistral
50	Random	2.09 ± 0.05	1.28 ± 0.11	1.47 ± 0.05	1.21 ± 0.09
	SSS-B	2.00 ± 0.04	1.14 ± 0.05	1.31 ± 0.07	1.01 ± 0.08
100	Random	1.80 ± 0.08	0.83 ± 0.05	1.09 ± 0.07	0.72 ± 0.05
	SSS-B	1.57 ± 0.08	0.70 ± 0.04	0.97 ± 0.05	0.70 ± 0.05

Table 4: Generalization results of SSS-B versus Random sampling across four LLMs (LLaMA3, LLaMA3-Instruct, Qwen2.5-Instruct, and Mistral) at sample sizes 50 and 100. SSS-B consistently reduces harmfulness scores compared to Random.

ior—are a key factor in maintaining model safety during fine-tuning.

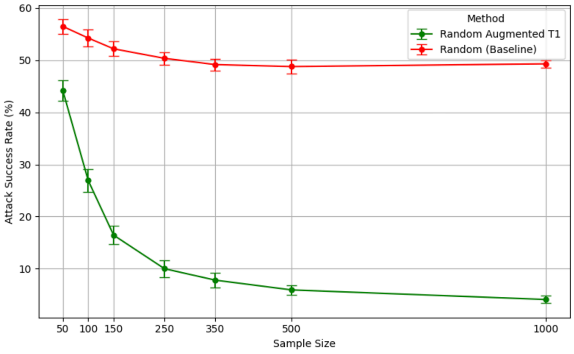

Figure 7: Attack sucess rate of randomly sampling compared to Beavertails augmentation baseline. *Random (Baseline)* only sample from safe pair in BeaverTail, while *Random Augmented T1* sample from our agumented dataset of only T1 data